

1 Parcurgeri

1. Explicați cum poate ajunge un vârf u al unui graf orientat într-un arbore de adâncime ce îl conține numai pe u , deși există în G atât muchii care intră în u cât și muchii care ies din u .
2. Se dă un graf neorientat conex $G = (V, E)$. Care este numărul maxim de muchii care pot fi șterse din G pentru ca noul graf $G' = (V, E')$ să respecte $d_G(1, i) = d_{G'}(1, i)$ (pt oricare $i, 1 \leq i \leq n$). Reformulare: Se dă un graf neorientat conex $G = (V, E)$ și un vârf s . Care este numărul minim de muchii ale unui graf parțial al lui G care conservă distanțele de la s la celelalte vârfuri?
3. Se dă un graf neorientat conex $G = (V, E)$. Se consideră parcurgerile DFS și BFS care pornesc din nodul 1, iar vecinii unui nod sunt considerați în ordine crescătoare. Ce proprietăți trebuie să respecte G pentru ca cele două parcurgeri să obțină același arbore parțial? - <https://www.hackerrank.com/contests/colocviu-af-123/challenges/problema-1-1-2>
4. Fie G_1 și G_2 două grafuri neorientate conexes cu același număr de muchii cu proprietatea că ordinea în care sunt parcurse vârfurile în BF pornind din vârfurile 1 este aceeași pentru ambele grafuri, la fel și în DF (vecinii unui vârf sunt parcurși în ordine crescătoare). Sunt G_1 și G_2 egale?
5. (Temă) Fie G_1 și G_2 două grafuri neorientate conexes cu același număr de muchii cu proprietatea că arborii BF și DF ai celor două grafuri sunt egali (vecinii unui vârf sunt parcurși în ordine crescătoare). Sunt G_1 și G_2 egale?
6. Elaborati un algoritm care determina dacă un graf neorientat $G=(V,E)$ dat contine sau nu un ciclu. Algoritmul trebuie sa ruleze în $O(n)$, $O(n)$ indiferent de m .
7. (Temă) Fie G un graf neorientat, s un vârf din G și k un număr natural pozitiv. Descrieți un algoritm de a parcurge toate vârfurile aflate la distanță mai mică sau egală cu k decât s fără a parcurge (descoperii, vizita) alte vârfuri. Justificați corectitudinea algoritmului propus.

2 Structuri UNION-FIND -aplicații. APCM

1. <https://www.infoarena.ro/problema/disjoint> și <https://cses.fi/problemset/task/1676/>
2. Aplicatie UNION-FIND - Se consideră n stații între care pot exista legături. Două stații conectate printr-o legatură pot comunica direct. Spunem ca doua statii comunica indirect daca exista statii intermediare prin intermediul carora pot comunica. Cele n statii formeaza in prezent k clustere, adica k grupuri disjuncte cu proprietatea ca oricare doua statii dintr-un cluster pot comunica direct sau indirect, si oricare doua statii din clustere diferite nu pot comunica. Intre statii se stabilesc noi legaturi de comunicare, date prin perechi (i,j) cu semnificatia statia i comunica direct cu statia j . Sa se citeasca aceste perechi si dupa fiecare citire sa se afiseze numarul de clustere existente in prezent (mai mult: si dimensiunea celui mai mare cluster - union by size in loc de union by rank).
3. Modificati algoritmul lui Kruskal astfel încât, date 2 muchii e_1, e_2 sa determine, un arbore parțial al lui G care contine cele 2 muchii. Daca exista mai multi astfel de arbori, se va afisa cel de cost

minim (conectare cu cost minim, dar cu muchii care trebuie obligatoriu construite).

4. a) Dați exemplu de graf care are exact doi apcm.

b) Fie G un graf conex ponderat cu ponderile muchiilor distincte. Arătați că există un unic apcm al lui G .